# SIAMESE ARCHITECTURES FOR LEARNING SENTENCE SIMILARITY

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND COMMUNICATIONS ENGINEERING,
THE GRADUATE SCHOOL OF FUNDAMENTAL SCIENCE
AND ENGINEERING
OF WASEDA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF ENGINEERING

July 16th 2021

Haoxiang Shi
5119FG19-7

Advisor: Prof. Tetsuya Sakai
Research guidance: Research on Information Access

## Abstract

This thesis presents two deep neural architectures which apply the Siamese Neural Network sharing model parameters for learning a semantic similarity metric between two sentences. In addition, two different similarity metrics (i.e., the Cosine Similarity and Manhattan similarity) are compared based on these two architecture. Our experiments in binary similarity classification for sentence pairs (in both English and Chinese) show that the proposed Siamese BERT architecture with Manhattan similarity achieves the best performance. More specifically, for the Chinese tasks, the Siamese BERT architecture with Manhattan similarity outperforms the baselines (i.e., the Siamese Long Short-Term Memory architecture and the Siamese Bidirectional Long Short-Term Memory architecture) and he Siamese CNN architecture in term of accuracy by 8.74, 8.75 and 0.07 points, respectively. In this case, the Siamese CNN architecture achieves almost the same performance with the Siamese BERT architecture, which suggests a strong advantage with a lighter structure.

# Contents

# Chapter 1

# Introduction

Measuring the similarity between words, sentences, paragraphs and documents is an important component in various tasks such as information retrieval, document clustering, word-sense disambiguation, automatic essay scoring, short answer grading, machine translation and text summarization. Traditional sentence similarity measurement is based on the edit distance, Jaccard index, and the bag-of-words models such as TF-IDF. These methods of learning sentence similarity are in fact based on the word level, which may not be sufficient. For example, there are two Chinese sentences as shown in Figure 1. The corresponding English translations are "How to buy LCD TVs." and "What kind of LCD TVs is good?". From the word level (i.e., character level in Chinese), the two sentences look the same, but they have totally different meaning at the sentence level. That is, we need sentence-level methods to capture the semantics of the sentences for sentence similarity measurement.

With the rapid development of machine learning, using neural network to learn representations of sentence-level meanings has been widely verified to be effective. The beginning of using neural network to learn sentence-level representations may be the Word2Vec from Google [1], which used a shallow structure to learn the vector-based representations of sentence level. However, using one neural architecture to learn two sentences in two steps may cause inconsistent representations. Hence, Siamese structures, which can learn two sentences at a time, are attractive alternatives. The Siamese architecture that can achieve state-of-the-art accuracy results in learning English sentence similarity is a Bidirectional Long Short-Term Memory (Bi-LSTM) based Siamese recurrent architecture [2].

In our preliminary study, we tested the effectiveness of a Siamese recurrent architecture for learning Chinese sentence similarities. However, this did not perform as well as what is reported by Neculoiu et al. [2]. Therefore, we tried different Siamese architectures including the convolutional one inspired from the image processing field [3] and Bidirectional Encoder Representations from Transformers (BERT) [4] one to implement sentence similarity learning. The results in binary similarity classification for sentence pairs show that the proposed

Siamese architectures outperform the Siamese recurrent architecture in learning accuracy. In addition, we consider two similarity metrics in each Siamese architecture, namely, the Manhattan similarity and the Cosine similarity. The results show that a Siamese architecture plus the Manhattan similarity performs better than other baselines for learning the similarity between two sentences.

Our contributions are as follows: (1) we verified Siamese BERT is the best architecture (except for the case of using cosine similarity for Chinese); (2) we verified that the Manhattan similarity can achieve better performance than other similarity metrics regardless of the learning architectures; (3) particularly, we verified that Siamese convolutional architecture is effective in learning Chinese sentence semantic similarity.

The structure of this thesis is as follows: in Chapter 2, we listed the current Siamese architectures; in Chapter 3, we illustrated our Saimese architecture by using convolutional neural network (CNN) and BERT, respectively; in Chapter 4, we showed the datasets and experiment setups; finally, in Chapter 5, we displayed the similarity accuracy learned by the proposed Saimese methods.

# Chapter 2

# Related Work

The Siamese network [5] is firstly proposed for non-linear metric learning with similarity information. It naturally learns representations that embody the invariance and selectivity desiderata through explicit information about similarity between pairs of objects. The Siamese architecture has since been widely used in vision applications. Specifically, the Siamese convolutional networks were used to learn complex similarity metrics for face verification [6] and dimensionality reduction on image features [7]. While in the natural language processing (NLP) field, the Convolutional Neural Network (CNN) has attracted more attentions since the successes in using CNN to do the traditional NLP tasks [8], and the availability of high-quality semantic word representations has been verified when using the CNN [1].

Recently, CNNs have been applied to matching sentences [9]. Although the work [9] has used the CNN to learn representations of two sentences, this is not a Siamese CNN architecture. Following this, the Siamese Long Short-Term Memory (LSTM) architecture was proposed for sentence similarity task using token level embedding [10]. Subsequently, a Siamese Bi-LSTM structure was proposed in order to improve the result of sentence similarity [2]. A Siamese CNN combines Bi-LSTM structure has been proposed for learning sentence similarity [11]. However, this architecture achieves lower accuracy than the independent Bi-LSTM structure. Also, Pontes et al. [11] did not give any comparisons between Siamese CNN architecture and Siamese Bi-LSTM architecture. Later, Siamese LSTM and Siamese Bi-LSTM were compared based on an English dataset [12].

# Chapter 3

# Proposed Methods

## 3.1 Siamese Architecture

The proposed Siamese architecture is depicted in Fig. 3.1. In the architecture, there are two exactly alike encoder structures that are used. The inputs of each encoder are the world-level (for English) or character-level (for Chinese) embeddings of a sentence, and the outputs of each encoder structure are the sentence level representations. Then, a similarity metric is used to compare the outputs of the two convolutional structures. The calculated similarity is the final output of the Siamese architecture.

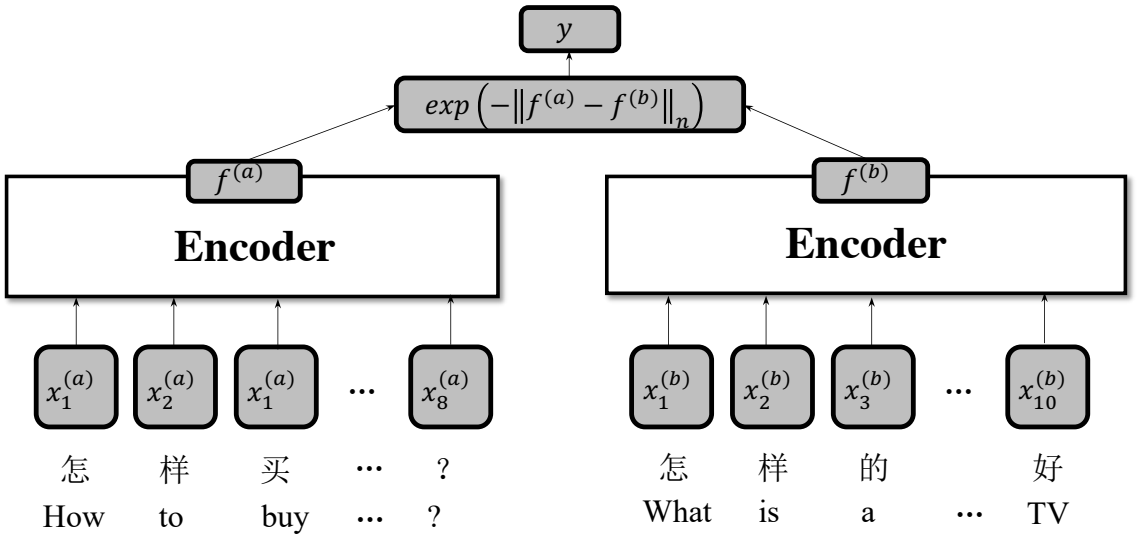


Fig. 3.1: A Siamese architecture.

The similarity depicted in Fig. 3.1 is the exponential negative norm of two learned representation vectors, which is defined as:

$$sim_{Man} = \exp\left(-\left\|f^{(a)} - f^{(b)}\right\|_n\right) \tag{3.1}$$

where in equation manhattan similarity $f^{(a)}$ and $f^{(b)}$ are the representations of the two sentences from the two encoder structures. If $n = 1$, the similarity is the Manhattan distance-based similarity or the Manhattan similarity for short. If $n = 2$, the similarity is then the Euclidean distance-based similarity or the Euclidean similarity for short. We have also tested the performance of the Siamese network with Euclidean similarity. The accuracy is around 50%, which means the Euclidean similarity does not work well with the Siamese architecture. Therefore, this result is not shown in Chapter 5. The similarity can also be replaced by the Cosine similarity.

$$sim_{Cos} = \frac{(f^{(a)} \cdot f^{(b)})}{\left\|f^{(a)}\right\| \cdot \left\|f^{(b)}\right\|} \tag{3.2}$$

After calculating the similarity, we then use the mean-square error (MSE) of the similarity and the label as the loss function. The gradients of the loss will be fed back to both encoder structures. In this way, the two encoder structures will share the same parameters, and then they can learn the representations of the two sentences with the same distribution. Based on a threshold of the similarity, we can then evaluate the accuracy after learning.

## 3.2 Alternative Encoder

We adopted CNN and BERT as the candidate encoder. Note that for the Siamese architecture, encoder can be various according to the effectiveness.

### 3.2.1 CNN Encoder

The specific CNN encoder is shown in Fig. 3.2. Within each CNN encoder structure, there are one fully connected layer after three repeated convolutional layers and max pooling layers. Note that the number of the repeated parts can be adjusted. However, we have also tested the six repeated structure, the accuracy did not show a significant improvement. That is to say, the 3-layer CNN is sufficient to show effectiveness with acceptable complexity. The kernel size of each convolutional layer is different. A higher convolutional layer is equipped with a larger kernel size. The fully connected layer then reduces the dimension of the learned representations from pooling layer. The learned output vector from the fully connected layer will be used to calculate the similarity then.

### 3.2.2 BERT Encoder

The specific BERT encoder is shown in Fig. 3.3. BERT basically can be regarded as a stack of multiple Transformer encoders [13], which is an encoder-decoder network that uses self-attention in the encoder and attention in the decoder. BERT has three embeddings: token embeddings, segment embeddings and positional embeddings. According to

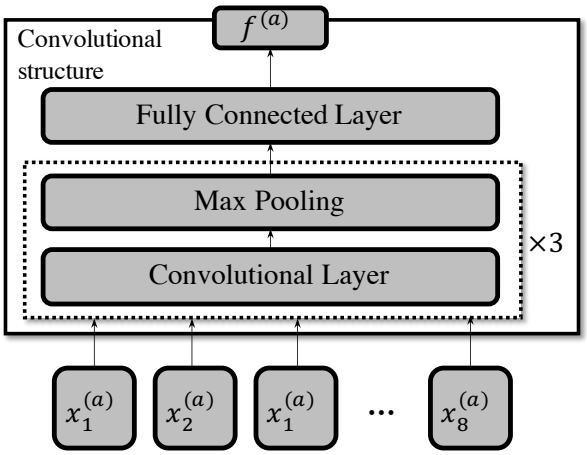


Fig. 3.2: A CNN encoder for the Siamese architecture.

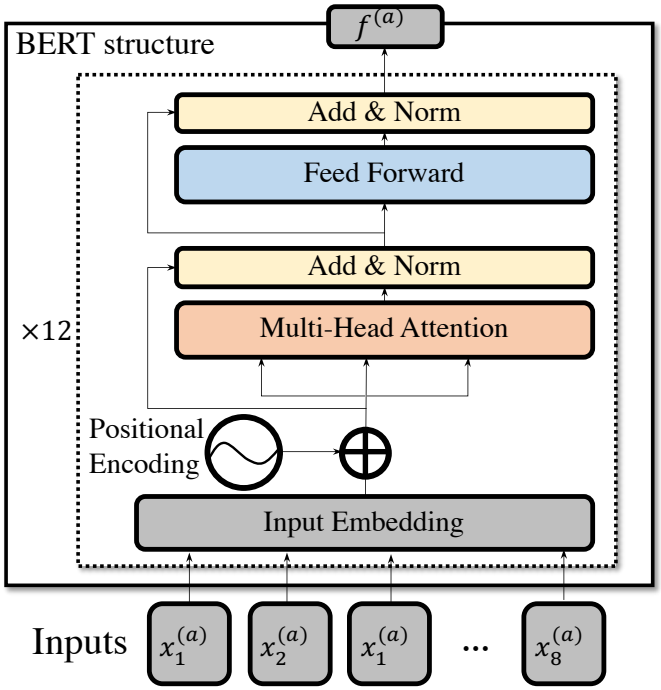


Fig. 3.3: A BERT encoder for the Siamese architecture.

Vaswani et al. [13], the novel positional embedding is adopted according to multiple trials. BERT varies in two scales, *BERT_Base* and *BERT_Large*. *BERT_Base* has 12 layers, and *BERT_Large* has 24 layers in the encoder stack. In addition to use the Transformer architecture with 6 encoder layers described in the original paper, the BERT architecture (Base and Large) also has a larger feedforward network (768 and 1024 hidden units, respectively) and more attention points (12 and 16, respectively). Moreover, it contains 512 hidden units and 8 attention heads. *BERT_Base* contains 110M parameters, while *BERT_Large* has 340M parameters.

Different from the CNN that can be customized, according to the aforementioned BERT encoder structure, the parameters of which is almost deterministic. Besides, BERT has the pre-training using two unsupervised tasks: masked language model and next sentence prediction. Thus, when we use BERT, we use the pre-trained BERT model (with parameters) and fine tune the model by the specific supervised task (i.e., the similarity learning in this thesis). Similar to the CNN encoder, the output of the BERT encoder is used for the similarity calculation.

# Chapter 4

# Experiments

## 4.1 Dataset

Our experiments are the binary similarity classification tasks for sentence pairs for both English and Chinese. In a specific dataset, a data record is always like <sentence 1, sentence 2, similarity> (i.e., 1 represents that the two sentences are similar and 0 represents that two sentences are dissimilar). Although obtaining a Chinese sentence similarity dataset is difficult, we found a dataset named LCQMC with even distribution of the labels (i.e., similar sentence pairs and dissimilarity sentence pairs occupy 50% and 50% of all dataset respectively) from Baidu. The format of the dataset is shown in Fig. 4.1. Punctuations of some sentences are omitted in the original data. This dataset consists 283,000 data records. We have chosen 250,000 data records as the training data, and 12,500 data records as the test data. Note that, LCQMC was open-source when we executed the experiment, however when the thesis publish, LCQMC will be no longer open. We also used the English dataset PAWS-X [14] to train and test the different Saimese architecture as the comparisons. In the PAWS-X, the data format is the same with LCQMC, and the task is also to learn the semantic similarity between two sentences. We used 49,401 data records in PAWS-X to train models and 2,000 to test. The dataset statistics are shown in Table. 4.1.

Table 4.1: Overview of datasets

| Datasets | # of class | Training/test size | Language |
|---|---|---|---|
| LCQMC | 2 | 250,000/12,500 | CH |
| PAWS-X | 2 | 49,401/2,000 | EN |

## 4.2 Baseline

From the aforementioned related works [2] [12], we have chosen two baselines: the Siamese Bi-LSTM architecture and the Siamese LSTM architecture. When performing the baselines

| Record | Sentence 1 | Sentence 2 | Label |
|---|---|---|---|
| 1 | 三星手机屏幕是不是最好的？<br>Is the screen of Samsung mobile phone the best or not? | 三星手机的屏幕是不是都很好<br>Are the screens of all kinds of Samsung mobile all good? | 0 |
| 2 | 广西桂林电子科技大学怎么样?<br>How about the Guilin University of Electronic Technology in Guangxi ? | 桂林电子科技大学怎么样<br>How about the Guilin University of Electronic Technology ? | 1 |
| 3 | 支付宝钱包怎么用<br>How to use Alipay? | 支付宝钱包怎么样<br>How about Alipay? | 0 |
| ..... | ..... | ..... | |

Fig. 4.1: The format of the LCQMC.

for English dataset, to improve the performance lower-bound of the Siamese LSTM and Bi-LSTM architectures, we introduced Glove [15] as the embedding. We did not choose a specific embedding for Chinese because that there is still no a well-recognized Chinese embedding library. Moreover, we also evaluated the two baselines, the Siamese BERT architecture and the Siamese CNN architecture with two different loss functions (i.e., the Manhattan similarity based MSE and the Cosine distance based MSE).

## 4.3 Setup

We used *BERT_Base* and adopted the original embeddings of BERT model for both Chinese and English. As for the Siamese CNN architecture, the kernel sizes of the three repeated convolutional layers are set as 3, 4 and 5. When using Siamese CNN architecture for English, we also adopted the Glove.

For each Siamese architecture with different encoder, we ran a total of 100 epochs. The batch size of Siamese CNN architecture and Siamese BERT architecture is 128 and 16, respectively. The difference in the batch size is a consideration of the computing power limitation. The Adam optimizer is used. During the optimization, we set the learning rate to be 0.001 for Siamese CNN and $2 \times e^{-5}$ for Siamese BERT.

Following previous work [2], we used accuracy as the evaluation metric. We then set the similarity threshold as 0.5. That is to say, if the calculated similarity is more than 0.5, the prediction is that the two sentences are similar. Conversely, the similarity less than 0.5 is decided as dissimilar.If the similarity is exactly 0.5, the result is excluded for calculating accuracy.

The specific information of computing resources we used for the experiment is shown in

Table. 4.2

Table 4.2: Environment of Experiments

| | |
|---|---|
| CPU | Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz |
| Memory Size | 32GB |
| GPU | Nvidia Geforce RTX 1080ti |
| GPU Memory Size | 11GB |

# Chapter 5

# Results and Discussions

We first, in Section 5.1 and Section 5.2, showed detailed learning loss and accuracy of each architecture for the Chinese sentence similarity learning task. Then, in Section 5.3, we overall compared the learning results of the Chinese and English dataset. Additionally, we gave a brief discussion for the results in Section 5.4.

## 5.1 Result of Siamese CNN

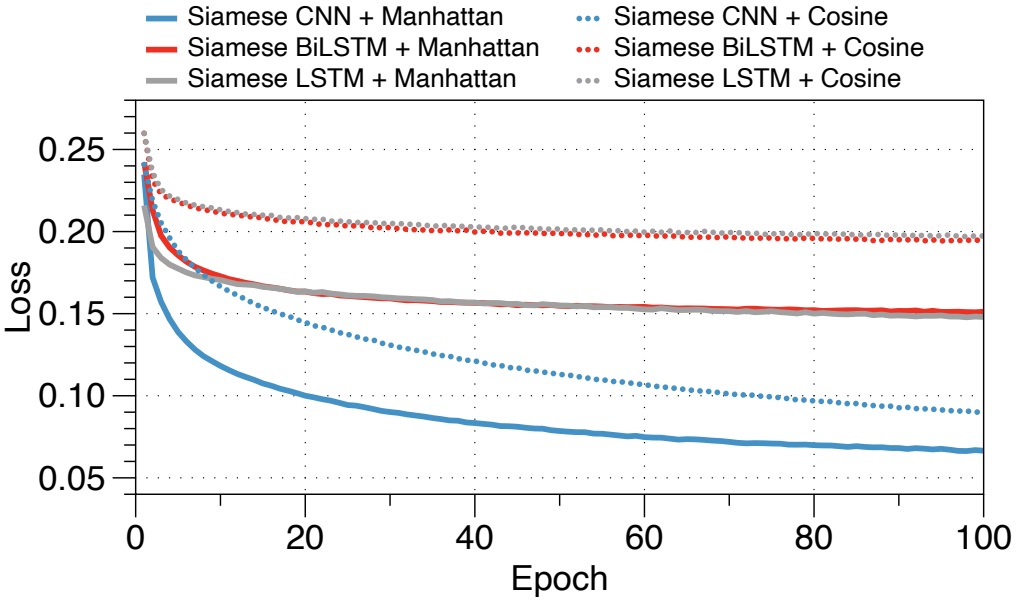


Fig. 5.1: The convergences and the losses of the Siamese CNN architectures and the baselines.

In Fig. 5.1, we compared the convergence speeds and losses of all combinations of the Siamese architectures (with CNN, Bi-LSTM and LSTM as the encoder) and the two loss functions (i.e., using Manhattan or Cosine similarity). The lines in different colors represent different Siamese architectures. The full lines are the losses using the Manhattan similarity, and the dotted lines are the losses using the Cosine similarity. It can be observed that no

matter what kind of the Siamese architecture is used, the Manhattan similarity based Siamese architectures converge fast. As for the final loss, the Siamese CNN architectures always achieve lower losses than the baselines. In the Siamese CNN architectures, the Manhattan similarity based Siamese architecture always gets a lower loss. As a result, the Siamese CNN architecture with the Manhattan similarity metric achieves the lowest loss. Regardless of the choice the similarity metric, the losses of the Siamese LSTM architecture and the Siamese Bi-LSTM architecture are similar.

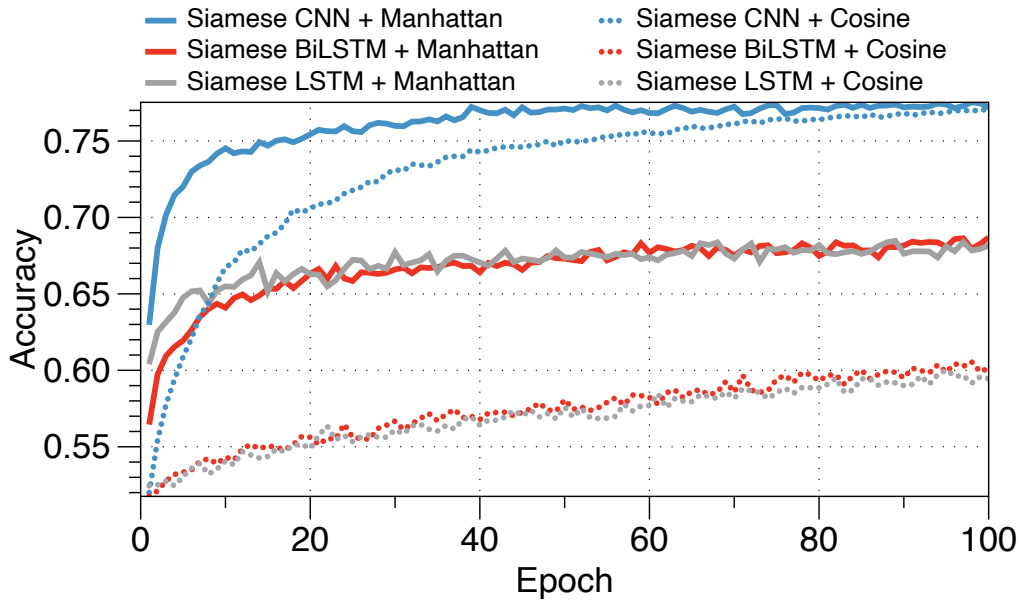


Fig. 5.2: The accuracies of the Siamese CNN architectures and the baselines.

Next, we evaluated the accuracy of all the combinations of the Siamese architectures (with CNN, Bi-LSTM and LSTM as the encoder) and the two loss functions (i.e., using Manhattan or Cosine similarity). The representation formats of different combinations are the same with Fig. 5.1. As shown in Fig. 5.2, it can be seen that the Siamese CNN architectures always achieve higher accuracy. In the Siamese convolutional architectures, the one with the Manhattan similarity metric always achieves higher accuracy. In summary, the Siamese CNN architecture with the Manhattan similarity metric can obtain the highest accuracy. The performances of the two baselines are not substantially different regardless of the similarity metric. The Siamese Bi-LSTM architecture shows a slight improvement of the accuracy comparing to the Siamese LSTM architecture.

## 5.2 Result of Siamese BERT

In Fig. 5.3, we additionally compared the convergence speeds and losses of all combinations of the three Siamese architectures (BERT, Bi-LSTM and LSTM) and the two loss

functions (i.e., using Manhattan or Cosine similarity). The representation of lines is the same as that in Fig. 5.1. It shows that the convergence is extremely fast when using Siamese BERT architecture. However, unlike the significant difference in Siamese CNN architecture, the convergence of using Manhattan similarity and Cosine similarity almost is equally fast. Moreover, we see that the final loss of the Cosine similarity-based Siamese BERT is just slightly lower than that of the Manhattan similarity-based Siamese BERT.

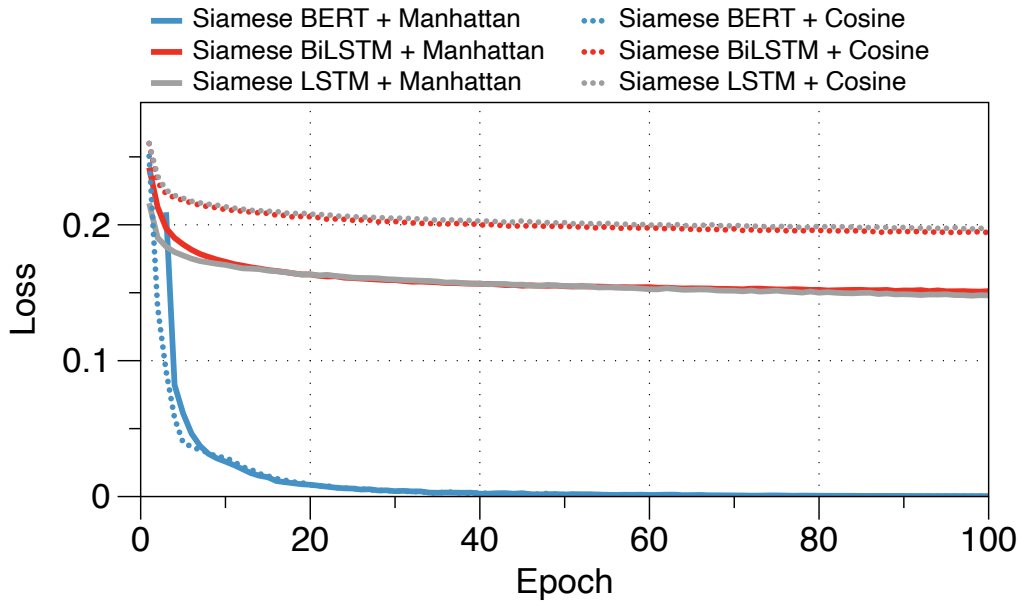


Fig. 5.3: The convergences and the losses of the Siamese BERT architectures and the baselines.

Similarly, we evaluated the accuracy of all the combinations of the Siamese architectures (with BERT, Bi-LSTM and LSTM as the encoder) and the two loss functions (i.e., using Manhattan or Cosine similarity). The representation formats of different combinations are the same with Fig. 5.1. In Fig. 5.4, it first can be seen that the accuracy of Siamese BERT grows significantly fast. The Manhattan similarity-based Siamese BERT architecture achieves the highest accuracy, however, the performance difference is not significant when compared to the Cosine similarity-based one. the one with the Manhattan similarity metric always achieves higher accuracy. We also can find a peak in the curve painted in blue solid line, where the accuracy is over 80%. It may suggest that when the epoch is set to be larger, the higher accuracy will be obtained by the Manhattan similarity-based Siamese BERT architecture.

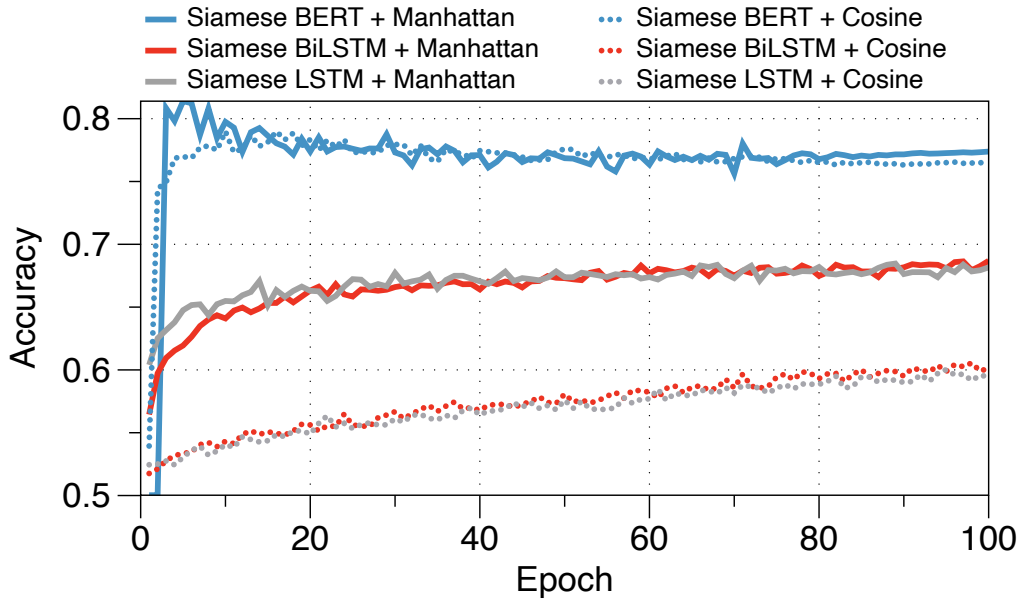


Fig. 5.4: The accuracies of the Siamese BERT architectures and the baselines.

## 5.3 Summary of the Results

We listed all the experimental results in the Table 1, including using LCQMC dataset and PAWS-X dataset. It can be observed that when using LCQMC dataset, the Siamese BERT architecture with Manhattan similarity achieves the best learning accuracy. However, when using Cosine similarity as the metric, the Siamese CNN even outperforms the Siamese BERT architecture. The performance difference between the Siamese CNN and Siamese BERT architecture is not significant. More specifically, with the metric of Manhattan similarity, the Siamese BERT architecture outperforms the Siamese CNN, the Siamese Bi-LSTM and the Siamese LSTM architecture by 0.07 points, 8.74 and 8.75 points, respectively. In addition, with the metric of Cosine similarity, our Siamese CNN architecture outperforms the Siamese BERT, Siamese Bi-LSTM and the Siamese LSTM architecture by 0.56, 16.50 and 17.03 points, respectively. It suggests that for a Chinese sentence similarity learning task, the Siamese CNN architecture is strongly effective. Note that as a CNN is much lighter (i.e., the scale of the parameters) than the BERT, it seems that the Siamese CNN is always the first candidate regardless of the metrics.

On the other hand, for the PAWS-X (the English dataset), the Siamese BERT architecture still shows powerful performance. The performance gap between Siamese BERT and other architectures is significantly wide, even when the baselines adopted the Glove embedding. However, the Siamese LSTM architecture and Siamese Bi-LSTM architecture outperform Siamese CNN architecture.

| Dataset | Architecture | Man. Similarity | Cos. Similarity |
|---|---|---|---|
| LCQMC | Sia. BERT architecture | **77.38** | 76.49 |
| | Sia. CNN architecture | 77.31 | **77.05** |
| | Sia. Bi-LSTM architecture | 68.64 | 60.55 |
| | Sia. LSTM architecture | 68.63 | 60.02 |
| PAWS-X | Sia. BERT architecture | **76.25** | **73.70** |
| | Sia. CNN architecture | 57.80 | 56.41 |
| | Sia. Bi-LSTM architecture | 67.75 | 69.20 |
| | Sia. LSTM architecture | 68.14 | 67.45 |

Table 5.1: Accuracy comparison for different architectures with Manhattan and Cosine similarities.

## 5.4 Discussion

Through result analysis, it can be found that Siamese BERT has high performance and wide generalization ability due to the advantage for the BERT encoder. When using the BERT as the encoder in the Siamese architecture, the metric seems to be not so essential. Note that such advantage is partially from the huge number of parameters. However, for a Chinese task, the Siamese CNN also shows significantly strong performance (as good as the BERT encoder). Nevertheless, the Siamese CNN has weak generalization ability, that is, it performs bad for the English task. The performance discrepancy of the Siamese CNN architecture between Chinese and English may because that part of the CNN can do character-level encoding for Chinese. This is also why in some Chinese language tasks [16] [17], a CNN-based character-level encoder is added before the word-level or sentence-level encoding.

In recent years, to find an advantage of using BERT is not a big news. However, we found that the CNN as an encoder in the Siamese architecture performs as strong as BERT for the Chinese task, which is an interesting discovery.

# Chapter 6

# Conclusion

In this thesis, we proposed a Siamese BERT and a Siamese CNN architecture for sentence similarity learning tasks. The experiment show several essential results:

- The Siamese BERT achieves the highest accuracy in the following cases: a Chinese task with Manhattan similarity as the metric, English tasks with both Manhattan similarity and Cosine similarity as the metric.
- The Siamese BERT and the Siamese CNN outperform the baselines when conducting the Chinese tasks.
- For the Chinese tasks, the performance of the Siamese CNN and the Siamese BERT architecture is almost the same; in the case of taking Cosine similarity as the metric, the Siamese CNN even slightly outperforms the Siamese BERT architecture.
- For the English tasks, in contrast, the Siamese CNN underperforms the baselines.

According to the aforementioned results, we can draw the following main conclusions:

- The Siamese BERT architecture has good performance and strong generalization ability because of the well-designed BERT architecture.
- The Siamese CNN architecture has a significant advantage in a Chinese sentence similarity learning task. It defeats the advantage from the BERT structure using less parameters.
- The Siamese CNN architecture is not good for an English sentence similarity learning task even using the Glove.

Additionally, Manhattan similarity metric always can help to achieve faster convergence and higher accuracy than any other similarity metric. We may also suggest that the Siamese architectures which are effective in English NLP tasks may not necessarily work well in Chinese NLP tasks. We should do more works for the language differences.

# Chapter 7

# Future Work

In the future, we will try to build and conduct experiments on Siamese Transformer [18] architecture. In addition, we will design more comparative experiment to try to explain the reason why same architecture performs differently for tasks in different languages. Moreover, we will use more sentence similarity corpus to pre-train the BERT and use the pre-trained BERT model to build the Siamese BERT architecture to further improve the performance.

# Acknowledgements

This research was undertaken with the support from The Real Sakai Laboratory [19], Waseda University. I would like to express my thanks of gratitude to members in The Real Sakai Laboratory, who helped me and gave me many useful pieces of advice for continuing the research. Also, I would like to thank Professor Tetsuya Sakai, who gave me this great opportunity and lots of supports to do research on this topic.

# References


[1] T. Mikolov, K. Chen I. Sutskever, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[2] P. Neculoiu, M. Versteegh, and M. Rotaru. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157, 2016.

[3] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, pages 148–157, 2015.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *Advances in Neural Information Processing Systems*, pages 737–744, 1993.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 539–546, 2005.

[7] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742, 2006.

[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 1993.

[9] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. *Advances in Neural Information Processing Sys-*

*tems*, pages 2042–2050, 2014.

[10] J. Mueller and A. Thyagarajan. Dimensionality reduction by learning an invariant mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[11] E. L. Pontes, S. Huet, A. C. Linhares, and J. M. Torres-Moreno. Predicting the semantic textual similarity with siamese cnn and lstm. arXiv:1810.10641, 2018.

[12] T. Ranasinghe, C. Orasan, and R. Mitkov. Semantic textual similarity with siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] Y. Yang, Y. Zhang, C. Tar, and J. Baldridge. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. arXiv:1908.11828, 2019.

[15] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[16] F. Dai and C. Zheng Cai. Glyph-aware embedding of chinese characters. arXiv:1709.00028, 2017, 2017.

[17] T. Su and H. Lee. Learning chinese word representations from glyphs of characters. arXiv:1708.04755, 2017.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[19] RSL. The sakai laboratory. Last updated: 26th March 2019.